H9SPAD: Secure Programming for Application Development

Module Code:	H9SPAD
Long Title	Secure Programming for Application Development APPROVED
Title	Secure Programming for Application Development
Module Level:	LEVEL 9
EQF Level:	7
EHEA Level:	Second Cycle
Credits:	5
Module Coordinator:	MICHAEL BRADFORD
Module Author:	Margarete Silva
Departments:	School of Computing
Specifications of the qualifications and experience required of staff	

Learning Outcomes

On successful completion of this module the learner will be able to:

#	Learning Outcome Description
LO1	Investigate and critically assess the impact of application security vulnerabilities on users of software products.
LO2	Investigate and critically assess the state of the art in the latest programming paradigms to create security controls that prevent common application security vulnerabilities.
LO3	Design and develop solutions that fix common software application security vulnerabilities.

Dependencies

Module Recommendations

No recommendations listed

Co-requisite Modules

No Co-requisite modules listed

Entry requirements	

H9SPAD: Secure Programming for Application Development

Module Content & Assessment

Indicative Content
Introduction (10%) · Security support for programming languages · Seven Pernicious Kingdoms · Native Code Exploitation Principles · Stack and Function Call
Principles of Secure Design (15%) · Least privilege and isolation · Fail-safe defaults · End-to-end security · Defence in depth · Security by design · Threat modelling · Tensions between security and other design goals
OS Exploit Mitigation (15%) · Data Execution Prevention/Non-Executable Stack/Heap · Return-to-libc and Return Oriented programming · Address Space Layout Randomisation · Heap Spray
Input Validation (20%) · Buffer Overflow Exploitation · Mitigating Controls: Canaries/Security Cookies/FORTIFY_SOURCE · Heap Overflows · Recommendations for Buffer Overflow · Command Injection · Recommendations for Command Injection · Format String Vulnerabilities · Recommendations for Format Strings · Integer Issues · Integer Overflow Exploitation · Recommendations for Integer Issues
Time and State (15%) · Race Conditions & TOCTOU · Classic Unix TOCTOU access()/fopen() · Recommendations for File system TOCTOU · Insecure Temporary File · Recommendations for Temporary Files
Code Quality and Review (25%) · Use-after-Free Issues · Double-Free Issues · NULL Pointer Dereference · Kernel-Land Exploitation · Type Confusion · Code Review · Code Analysis · Scanning and Assessment Tools · Automated Security Testing · Defensive Coding · Frameworks for Coding

Assessment Breakdown	%
Coursework	100.00%

Assessments

Full Time

Coursework

Assessment Type:	Continuous Assessment	**% of total:**	60
Assessment Date:	n/a	**Outcome addressed:**	1,2
Non-Marked:	No		

Assessment Description:
Practical work will be conducted throughout the semester to assess the learner’s evaluation skills in terms of secure design strategies and secure application development.

Assessment Type:	Project	**% of total:**	40
Assessment Date:	n/a	**Outcome addressed:**	2,3
Non-Marked:	No		

Assessment Description:
Students are required to complete a practical where they find and fix security vulnerabilities in a software application.

No End of Module Assessment

No Workplace Assessment

Reassessment Requirement

Coursework Only
This module is reassessed solely on the basis of re-submitted coursework. There is no repeat written examination.

Module Workload

Module Target Workload Hours 0 Hours

Workload: Full Time				
Workload Type	*Workload Description*	*Hours*	*Frequency*	*Average Weekly Learner Workload*
Lecture	No Description	1	Every Week	1.00
Tutorial	No Description	2	Every Week	2.00
Independent Learning	No Description	7.5	Every Week	7.50
			Total Weekly Contact Hours	3.00

Module Resources

Recommended Book Resources

R. C. Seacord. (2014), Secure Coding in C and C++, 2nd Edition. Addison-Wesley Professional.

J. C. Foster. (2005), Buffer Overflow Attacks: Detect, Exploit, Prevent, Syngress Press.

Supplementary Book Resources

D. LeBlank, M. Howard. (2004), Writing Secure Code, 2nd Edition. Microsoft Press.

J. Viega. (2003), Secure Programming Cookbook for C and C++: Recipes for Cryptography, Authentication, Input Validation & More, O'Reilly Media.

This module does not have any article/paper resources

Other Resources

[Website], SEI CERT C++ Coding Standard.
https://www.securecoding.cert.org/conflu ence/pages/viewpage.action?pageId=637

[Website], Protostar.
https://exploit-exercises.com/protostar/

[Website], Fusion.
https://exploit-exercises.com/fusion/

Discussion Note:	